Exhibit K8

2 of 2

SECTION 5: KEY MANAGEMENT

The proper management of cryptographic keys is essential to the effective use of cryptography for security. Keys are analogous to the combination of a safe. If a safe combination becomes known by an adversary, that safe provides no security against penetration by that adversary. Similarly, poor key management may easily compromise strong algorithms. Ultimately, the security of information protected by cryptography directly depends on the strength of the keys, the effectiveness of mechanisms and protocols associated with keys, and the protection afforded to the keys themselves. All keys need to be protected against modification (i.e., their integrity needs to be preserved), and secret and private keys (i.e., keys used by symmetric and asymmetric algorithms, respectively) need to be protected against unauthorized disclosure (i.e., their confidentiality needs to be maintained).

Key management provides the foundation for the secure generation, storage, distribution/establishment, use and destruction of keys, and is essential at all phases of a key's life. If a strong algorithm is used to encrypt data using keys that are properly generated, then the protection of that data can subsequently be reduced to just protecting the keys, i.e. the security of information protected by cryptography directly depends on the protection afforded the keys. Therefore, a Cryptographic Key Management System (CKMS) is required for managing the keys.

5.1 General Key Management Guidance

Several publications have been developed to provide general key-management guidance: SP 800-57 (see Section 5.1.1), FIPS 140 (see Section 5.1.2), and SP 800-131A (see Section 5.1.3).

5.1.1 Recommendation for Key Management

SP 800-57[53] provides general guidance on the management of cryptographic keys: their generation, use, and eventual destruction. Related topics, such as algorithm selection and appropriate key size, and cryptographic policy are also included in SP 800-57, which consists of three parts:

- SP 800-57, Part 1, *General Guidance*, contains basic key-management guidance, including:
  - The protection required for keying material;
  - Key life-cycle responsibilities;
  - Key backup, archiving and recovery;
  - Changing keys;
  - Cryptoperiods (i.e., the appropriate lengths of time that keys are to be used);
  - Accountability and auditing;

[53] SP 800-57, *Recommendation for Key Management*.

- Contingency planning; and
- Key compromise recovery (e.g., by generating new keys).

Federal agencies have a variety of information that they have determined to require cryptographic protection; the sensitivity of the information and the periods of time that the protection is required also vary. To this end, NIST has established four security strengths for the protection of information: 112, 128, 192 and 256 bits[54]. These security strengths have been assigned to the **approved** cryptographic algorithms and key sizes, and dates have been projected during which the use of these algorithms and key sizes is anticipated to be secure. For further information, see SP 800-131A.

Agencies need to determine the length of time that cryptographic protection is required before selecting an algorithm and key size with the appropriate security strength.

Note that SP 800-57, Part 1 will be updated if the guidance provided therein is no longer valid (e.g., an algorithm no longer provides adequate security).

- SP 800-57, Part 2, *Best Practices for Key Management Organization*, contains:
  - A generic key-management infrastructure,
  - Guidance for the development of organizational key-management policy statements and key-management practices statements,
  - An identification of key-management information that needs to be incorporated into security plans for general support systems and major applications that employ cryptography, and
  - An identification of key-management information that needs to be documented for all federal applications of cryptography.
- SP 800-57, Part 3, *Application-Specific Key Management Guidance*, addresses the key management issues associated with currently available cryptographic mechanisms, such as the Public Key infrastructure (PKI), Internet Protocol Security (IPsec), the Transport Layer Security protocol (TLS), Secure/Multipart Internet Mail Extensions (S/MIME), Kerberos, Over-the-Air Rekeying (OTAR), Domain Name System Security Extensions (DNSSEC), Encrypted File Systems and the Secure Shell (SSH) protocol.

  Specific guidance is provided regarding:
  - The recommended and/or allowable algorithm suites and key sizes,
  - Recommendations for the use of the mechanism in its current form for the protections of federal government information, and

[54] A fifth security strength (i.e., 80 bits of security) was acceptable for applying cryptographic protection (e.g., encryption) prior to 2014. However, this strength is not adequate.

- Security considerations that may affect the effectiveness of key-management processes and the cryptographic mechanisms using keys that are generated and managed by those key-management processes.

Note that in the case of TLS, a reference is provided to a separate publication – SP 800-52[55] – that provides extensive details for using TLS.

New key-management techniques and mechanisms are constantly being developed, and existing key-management mechanisms and techniques are constantly being refined. While the security-guidance information contained in Part 3 will be updated as mechanisms and techniques evolve, new products and technical specifications can always be expected that are not reflected in the current version of the document. Therefore, the context provided may include status information, such as version numbers or implementation status at the time that the document was last revised.

5.1.2 Security Requirements for Cryptographic Modules

FIPS 140 provides minimum security requirements for cryptographic modules that embody or support cryptography in federal information systems. A cryptographic module performs the actual cryptographic computations for a security system protecting sensitive information. The security requirements cover areas related to the secure design and implementation of a cryptographic module, including the module specification; cryptographic module ports and interfaces; roles, services and authentication; finite-state models; physical security; the operational environment; cryptographic key management; electromagnetic interference/electromagnetic compatibility (EMI/EMC); self-tests; design assurance; and the mitigation of attacks.

FIPS 140 is applicable to all federal agencies that use cryptography to protect sensitive information in computer and telecommunications systems. Further information about FIPS 140 and the validation of cryptographic modules is available at http://csrc.nist.gov/groups/STM/cmvp/index.html.

5.1.3 Transitions to New Cryptographic Algorithms and Key Lengths

With the development and publication of SP 800-57, Part 1, NIST provided recommendations for transitioning to new cryptographic algorithms and key lengths because of algorithm breaks or the availability of more powerful computers that could be used to efficiently search for cryptographic keys. SP 800-131A was developed to provide more specific guidance for such transitions. Each algorithm and service is addressed in

[55] SP 800-52, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*.

SP 800-131A, indicating whether its use is acceptable[56], deprecated[57], restricted[58], allowed only for legacy applications[59], or disallowed.

Note that SP 800-131A will be updated if the guidance provided therein is no longer valid (e.g., an algorithm no longer provides adequate security).

5.2 Cryptographic Key Management Systems

Several publications have been developed for the development of key-management systems: SP 800-130[60] (see Section 5.2.1), SP 800-152[61] (see Section 5.2.2) and documents relating to the Public Key Infrastructure used for asymmetric-key cryptography (see Section 5.2.3).

A Cryptographic Key Management System (CKMS) includes policies, procedures, components and devices that are used to protect, manage and distribute cryptographic keys and associated information (called metadata). A CKMS includes all devices or subsystems that can access a key or its metadata. The devices could be computers, cell phones, tablets, or other smart devices, such as cars, alarm systems, or refrigerators.

5.2.1 Key Management Framework

SP 800-130 contains topics that should be considered by a CKMS designer when developing a CKMS design specification. Topics include security policies, cryptographic keys and metadata, interoperability and transitioning, security controls, testing and system assurances, disaster recovery, and security assessments.

For each topic, SP 800-130 specifies one or more documentation requirements that need to be addressed by the designer. SP 800-130 is intended to assist in:

- The definition of the CKMS design by requiring the specification of significant CKMS capabilities,
- Encouraging CKMS designers to consider the factors needed in a comprehensive CKMS,
- Logically comparing different CKMSs and their capabilities,
- Performing security assessments by requiring the specification of implemented and supported CKMS capabilities, and

[56] No security risk is known at present.

[57] The use of the algorithm and key length is allowed, but the user must accept some risk.

[58] The use of the algorithm is discouraged, and there are additional restrictions required for use.

[59] The algorithm and key length may be used to process already-protected information, but there may be a risk in doing so.

[60] SP 800-130, *A Framework for Designing Cryptographic Key Management Systems*.

[61] SP 800-152, *A Profile for U. S. Federal Cryptographic Key Management Systems (CKMS)*.

- Forming the basis for the development of Profiles that specify the specific requirements for the CKMS to be used by an organization.

5.2.2 Key Management System Profile

SP 800-152 contains requirements for the design, implementation, procurement, installation, configuration, management, operation and use of a CKMS by and for U.S. federal organizations and their contractors. The Profile is based on SP 800-130 (see Section 5.2.1). SP 800-152 specifies requirements, makes recommendations for federal organizations having special security needs and desiring to augment the base security and key-management services, and suggests additional features that may be desirable to implement and use.

In addition to providing design requirements to be incorporated into a CKMS design, SP 800-152 provides requirements for a Federal CKMS (FCKMS) to be operated by a service provider that may be a federal agency or a third party operating an FCKMS under contract for one or more federal agencies and their contractors.

This Profile is intended to:

- Assist CKMS designers and implementers in supporting appropriate cryptographic algorithms and keys, selecting the metadata associated with the keys, and selecting protocols for protecting sensitive U.S. federal computing applications and data;
- Establish requirements for testing, procurement, installation, configuration, administration, operation, maintenance and usage of the FCKMS;
- Facilitate an easy comparison of one CKMS with another by analyzing their designs and implementations in order to understand how each meets the Framework and Profile requirements; and
- Assist in understanding what is needed to evaluate, procure, install, configure, administer, operate, and use an FCKMS that manages the cryptographic keys that protect sensitive and valuable data obtained, processed, stored, and used by U.S. federal organizations and their contractors.

5.2.3 Public Key Infrastructure

A PKI is a security infrastructure that creates and manages public-key certificates to facilitate the use of public-key (i.e., asymmetric-key) cryptography. To achieve this goal, a PKI needs to perform two basic tasks:

1. Generate and provide public key certificates that bind public keys to the identifier associated with the owner of the corresponding private key[62] and to other required information *after* validating the accuracy of the information to be bound, and

[62] The identifier could be the true identity of the owner, or could be an alias or a pseudonym used to represent the owner.

2. Maintain and provide certificate-status information for unexpired and revoked certificates.

Two types of certificates are commonly used: certificates used to provide the public keys that are used to verify digital signatures, and certificates used to provide the public keys used for key management (i.e., key establishment). Each certificate associated with digital signatures provides the public keys of one of the three digital-signature algorithms approved in FIPS 186: DSA, ECDSA or RSA (see Section 3.3). Certificates that convey the public keys to be used for key establishment may be of two types: those that provide a key-agreement public key (see Section 5.3.3), and those that provide a key-transport public key (see Section 5.3.4). Key-usage bits in a certificate indicate the purpose for which the public key is intended to be used.

As discussed in Section 3.3, public keys can be made available to anyone. However, a private key must be maintained under the exclusive control of the owner of that private key[63] (i.e., the user that is authorized to use the private key).

- If a private key that is used to generate digital signatures is lost, the owner can no longer generate digital signatures; some policies may permit users to maintain backup copies of the private key for continuity of operations, but this is not encouraged, so an alternative is to simply generate new key pairs and certificates.
- If the private key used to generate digital signatures is compromised, relying parties can no longer trust the digital signatures generated using that private key (e.g., someone may be using the signature to provide false information).
- If a private key used for key establishment is lost (e.g., a key used for key transport or key agreement), then further key establishment processes cannot be accomplished until the key is recovered or replaced; if the key is needed to recover data protected by the key, then that data is lost unless the key can be recovered. For example, if the key is used to transport a decryption key for encrypted data, and the key is lost, then the encrypted data cannot be decrypted. To ensure that access to critical data is not lost, PKIs often backup the private key-establishment key for possible recovery.
- If a private key used for key establishment is compromised, then any transactions involving that key cannot be trusted (e.g., someone other than the true owner of the private key may be attempting to enter into a supposedly "secure" transaction for some illicit purpose).

5.2.3.1 PKI Components, Relying Parties and Their Responsibilities

For scalability, PKIs are usually implemented with a set of complementary components, each focused on specific aspects of the PKI process. The main PKI tasks are assigned to

[63] An exception could be some other trusted entity, such as the owner's organization. In these cases, the organization could be considered to be the *real* owner of the key.

the following logical components; other components are also used to support the PKI, but are not discussed here (see SP 800-32[64] for further discussion):

- *Certification authorities* (CAs) generate certificates and certificate-status information, and
- *Registration authorities* (RAs) verify the identity of users applying for a certificate[65] and authenticate other information to be included in the certificate.

In general, a PKI operates as follows:

1. An entity applies to an RA to request a certificate.
2. The RA verifies the identity of the applicant, and 2) verifies the information to be inserted in the certificate.
3. If the checks made by the RA in step 2 indicate that the information to be inserted in the certificate is valid, then the RA sends the public key and other relevant information to the CA to request that a certificate be generated.
4. Upon receiving the certificate request from the RA, the CA creates a digital certificate, returns the certificate to the RA and deposits the certificate in a repository.
5. When a relying party interacts with another entity that has a public-key certificate, the relying party needs to obtain the other entity's certificate, either directly or from the CA's repository. After acquiring the certificate, the relying entity verifies the signature on the certificate. Assuming that the certificate is "good," then the relying party can proceed safely with its interaction with the certificate's owner.

Most of the interaction involved with using a certificate is transparent to the user. However, a user or a system administrator may be responsible for obtaining and installing a certificate. Thereafter, an application (e.g., a browser) uses the certificate to interact with other entities, and the user may not be aware of these actions. An exception might be when a certificate has expired or been revoked, in which case a message may be displayed to indicate this status.

Certificates expire at a predetermined time unless revoked prior to the expiration date. Certificates can be revoked for a variety of reasons, including the compromise of the private key corresponding to the public key in the certificate, or the owner of the certificate leaving the organization. When a certificate has been revoked, a system will quite often display the certificate-revocation message and perhaps include the reason for the revocation. Depending on the application implementation and the revocation reason, the application could disallow further actions, or could allow the user to indicate whether to ignore the warning and continue operations, or to simply discontinue operations. This warning must not be taken lightly. Ignoring the warning means that the user is accepting the risks associated with doing so. For example, if a warning indicates a compromised

[64] SP 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*.

[65] The certificate could be for the user or for a device for which the user is authorized to obtain a certificate.

digital signature certificate, there is a possibility that someone other than the claimed owner of the certificate actually used the private key corresponding to the public key to sign data. Depending on the data, it may not be prudent to ignore the warning. A user should consult with his organization to determine how to respond to this warning.

5.2.3.2 Basic Certificate Verification Process

A PKI consists of at least one CA with its subscribers, as shown in Figure 5. Each of the subscribers (e.g., User 1, User 2 and User 3) obtains a certificate containing their public key and other information, which is signed by their CA. All CA subscribers are provided with the public key of the CA.

As a basic example of how this works, suppose that User 3 signs a document and sends it to User 1, who needs to verify the contents and source of the signed document. This is accomplished as follows:

1. User 1 obtains the certificate containing the public key that corresponds to the private key used to sign the document, i.e., User 1 obtains User 3's certificate. Either User 3 supplies that certificate, or the certificate is obtained from some other source, e.g., the CA.
2. User 1 verifies User 3's certificate using the CA's public key.
3. User 1 then employs the public key in User 3's certificate to verify the signature on the document received from User 3. If the signature is successfully verified, then User 1 knows that User 3 generated the signature, and no unauthorized modifications were made to the document after the signature was generated.

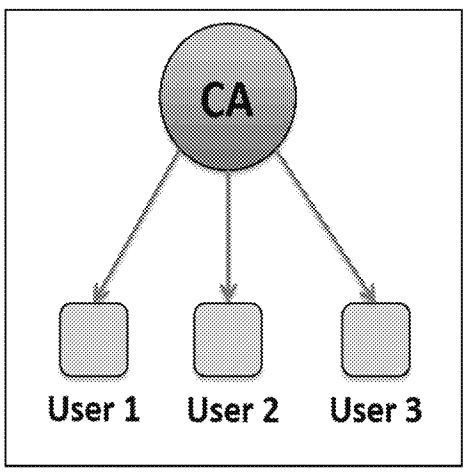

Figure 5: Basic Certificate Verification Example

Note that other more-complicated scenarios exist when users subscribing to different CAs need to interact using CAs that have cross certified by signing a certificate for each other.

5.2.3.3 CA Certificate Policies and Certificate Practice Statements

Each CA has a Certificate Policy and a Certificate Practices Statement. As defined by ITU[66] Recommendation X.509, a Certificate Policy (CP) is "a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements." The CP defines the expectations and requirements of the relying party community that will trust the certificates issued by the CAs using that policy. A CP addresses such issues as key generation and storage;

[66] International Telecommunication Union.

certificate generation; key escrow[67] and recovery; certificate status services, including Certificate Revocation List (CRL) generation and distribution; and system management functions, such as security audits, configuration management, and archiving.

A Certification Practice Statement (CPS) describes how a specific CA issues and manages public-key certificates. The CPS is derived from the applicable CP for the community or application in which the CA participates.

A Federal Public Key Infrastructure (FPKI) has been established for use by the Federal Government (see Section 5.2.3.4 for further information).

DRAFT NISTIR 7924[68] identifies a baseline set of security controls and practices to support the secure issuance of certificates. NISTIR 7924 is designed to be used as a template and guide for writing a CP for a specific community, or a CPS for a specific CA.

5.2.3.4 Federal Public Key Infrastructure

A Federal Public Key Infrastructure (FPKI) provides the Federal Government with a common infrastructure to administer digital certificates and public-private key pairs. The network portion of the FPKI (commonly referred to as the "Bridge") consists of "Principal CAs" designated by various agencies. Each CA within the bridge is cross-certified with every other CA within the bridge, thus establishing a conduit for trust relationships among all CAs within the FPKI. Each Principal CA may also be associated with other CAs that are not part of the bridge. For more information about the FPKI, including its certificate policy and certificate practices statement, see http://www.idmanagement.gov/federal-public-key-infrastructure.

5.3 Key Establishment

Key establishment is the means by which keys are generated and provided to the entities that are authorized to use them. An entity may be a person, organization, device or process. Scenarios for which key establishment could be performed include the following:

- A single entity could generate a key (see Section 5.3.1) and use it without providing it to other entities (e.g., for protecting locally stored data),
- A key could be derived from a key that is already shared between two or more entities (see Section 5.3.2),
- Two entities could generate a key using contributions (i.e., data) from each entity using an automated protocol that incorporates a key-agreement scheme (see Section 5.3.3), or

[67] Saving a key or information that allows the key to be reconstructed so that the key can be recovered if ever needed (e.g., because of being lost or corrupted).

[68] NISTIR 7924, *Reference Certificate Policy (Second Draft)*.

- A single entity could generate a key and provide it to one or more other entities, either by a manual means (e.g., a courier or a face-to-face meeting, with the key in either printed or electronic form, such as on a flash drive) or using automated protocols that incorporate a key-transport scheme (see Sections 5.3.4 and 5.3.5).

5.3.1 Key Generation

Cryptographic keys are required by most cryptographic algorithms, the exception being hash functions when not used as a component of another cryptographic process (e.g., HMAC). SP 800-133[69] discusses the generation of the keys to be used with the **approved** cryptographic algorithms.

All keys must be based directly or indirectly on the output of an **approved** Random Bit Generator (RBG) and must be generated within FIPS 140-compliant cryptographic modules (see FIPS 140). Any random value required by the module must be generated within a cryptographic module.

SP 800-133 provides guidance on generating a key directly from an RBG, and references other publications for additional information required for the generation of keys for specific algorithms:

- FIPS 186 provides rules for the generation of the key pairs to be used for the generation of digital signatures,
- SP 800-108 provides methods for the generation of keys from an already-shared key (see Section 5.3.2),
- SP 800-56A specifies the rules for the generation of key pairs for Diffie-Hellman and MQV key-agreement schemes (see Section 5.3.3),
- SP 800-56B specifies the rules for the generation of key pairs for RSA key-agreement and key-transport schemes (see Sections 5.3.3 and 5.3.4, respectively), and
- SP 800-132 specifies the rules for the generation of keys from passwords (see Section 5.3.6).

5.3.2 Key Derivation

Key derivation is concerned with the generation of a key from secret information, although non-secret information may also be used in the generation process in addition to the secret information. Typically, the secret information is shared among entities that need to derive the same key for subsequent interactions. The secret information could be a key that is already shared between the entities (i.e., a pre-shared key), or could be a shared secret that is derived during a key-agreement scheme (see Section 5.3.3).

SP 800-108[70] specifies several key-derivation functions that use pre-shared keys. A pre-shared key could have been

[69] SP 800-133, *Recommendation for Cryptographic Key Generation*.

[70] SP 800-108, *Recommendation for Key Derivation Using Pseudorandom Functions*.

- Generated by one entity and provided to one or more other entities by some manual means (e.g., a courier or face-to-face meeting),
- Agreed upon by the entities using an automated key-agreement scheme (see Section 5.3.3), or
- Generated by one entity and provided to another entity using an automated key-transport scheme (see Sections 5.3.4 and 5.3.5).

SP 800-56A, SP 800-56B and SP 800-56C[71] provide methods for deriving keys from the shared secrets generated during key agreement (see Section 5.3.3). SP 800-56A and SP 800-56 B specify two key-derivation methods for this purpose, and refer to SP 800-56C and SP 800-135[72] for additional **approved** methods[73].

5.3.3 Key Agreement

Key agreement is a key-establishment procedure in which the resultant keying material is a function of information contributed by all participants in the key-agreement process so that no participant can predetermine the value of the resulting keying material independently of the contributions of the other participants. Key agreement is usually performed using automated protocols.

SP 800-56A and SP 800-56B provide several automated pair-wise key-agreement schemes, i.e., key-agreement schemes involving two parties. For each scheme, a shared secret is generated, and keying material is derived from the shared secret using a key-derivation method specified or approved by reference in SP 800-56A, SP 800-56B or SP 800-56C.

SP 800-56A and SP 800-56B include variations of key-agreement schemes, differing in the number of keys used and whether the keys are long term (i.e., static) or an ephemeral value (e.g., a nonce or a short-term key pair). The key-agreement schemes have two participating entities: an initiator and a responder.

[71] SP 800-56C, *Recommendation for Key Derivation through Extraction-then-Expansion*.

[72] SP 800-135, *Recommendation for Existing Application-Specific Key Derivation Functions*.

[73] Note that a modification is in progress to move the KDF specifications and references in SP 800-56A and SP 800-56B to SP 800-56C.

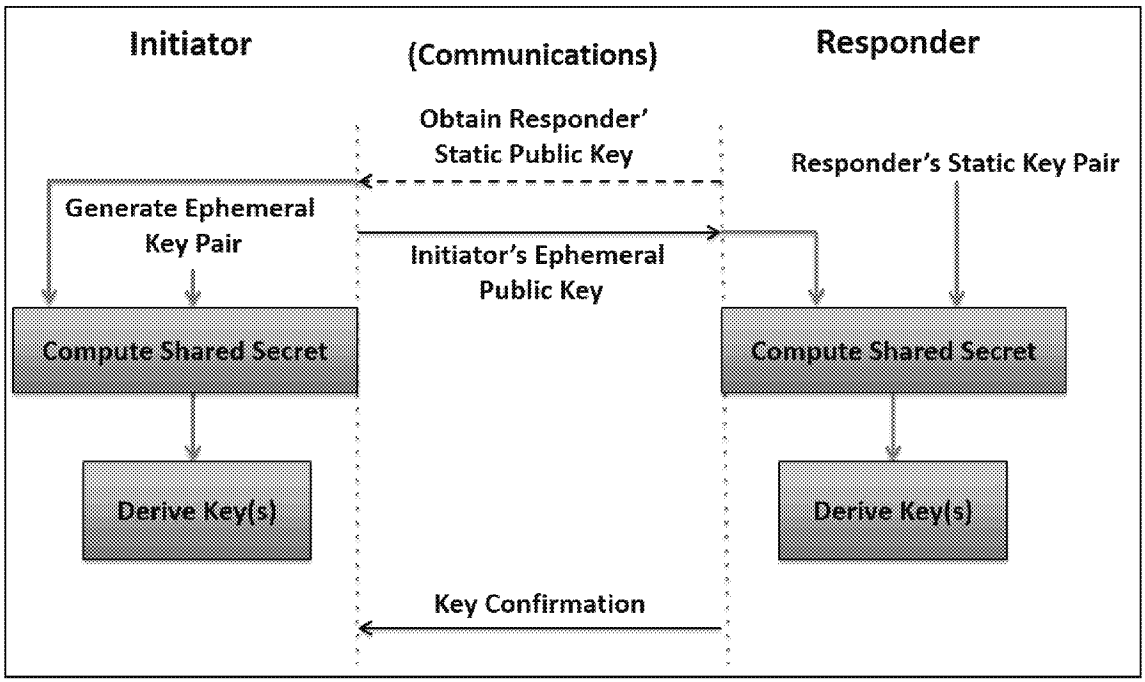

Figure 6: Key Agreement Example

Figure 6 provides an example of a key-agreement scheme where the responder uses a static key pair during the scheme, and the initiator uses an ephemeral key pair. Note that other key-agreement schemes may use other arrangements of key pairs (e.g., each party could use a static key pair or each party could use an ephemeral key pair). In the example provided in the figure above, the responder's private key is retained by the responder (who is the owner of the key pair), but the responder's public key may be provided to anyone. In this example, the public key is provided to the initiator:

1. The initiator obtains the responder's public key (e.g., from a CA or directly from the responder); for this scheme, this public key is the responder's contribution to the key-agreement process.
2. The initiator then generates a short-term key pair (i.e., an ephemeral key pair), and sends the ephemeral public key to the responder, retaining the ephemeral private key. The ephemeral public key is the initiator's contribution to the key-agreement process for this scheme.
3. Both parties use their own key pair and the other party's public key to generate a shared secret.
4. Both parties then use their copy of the shared secret to derive one or more keys that are (hopefully) identical.

Key confirmation is an optional, but highly recommended, step that provides assurance that both parties now have the same (identical) key(s), and is shown in Figure 6 for the case that the initiator receives key confirmation from the responder. See SP 800-56A and SP 800-56B for further information.

SP 800-56A specifies Diffie-Hellman (DH) and MQV key-agreement schemes using finite field or elliptic curve mathematics and asymmetric key pairs to generate the shared secret, and SP 800-56B specifies two RSA key-agreement schemes. SP 800-56A and SP 800-56B also provide an analysis of the merits of each key-agreement scheme.

5.3.4 Key Transport

Key transport is a method whereby one party (the sender) generates a key and distributes it to one or more other parties (the receiver(s)). Key transport could be accomplished using manual methods (e.g., using a courier) or performed using automated protocols. SP 800-56A and SP 800-56B provide automated pair-wise key-transport schemes, and an analysis of the merits of each key-transport scheme.

5.3.4.1 SP 800-56A Key Transport

SP 800-56A specifies a key-transport method whereby a key-establishment transaction includes both a key-agreement process and a key-wrapping process. Key wrapping is a process that provides both confidentiality and integrity protection for keying material using a symmetric-key algorithm (see Section 5.3.5 for further information about key wrapping).

During the transaction, the key generated during the key-agreement part of the transaction is used as a key-wrapping key with a symmetric-key algorithm (e.g., AES) by the sending party to wrap a key to be sent to the other party (the receiver). Note that the sender can be either the initiator or the responder in the key-agreement process.

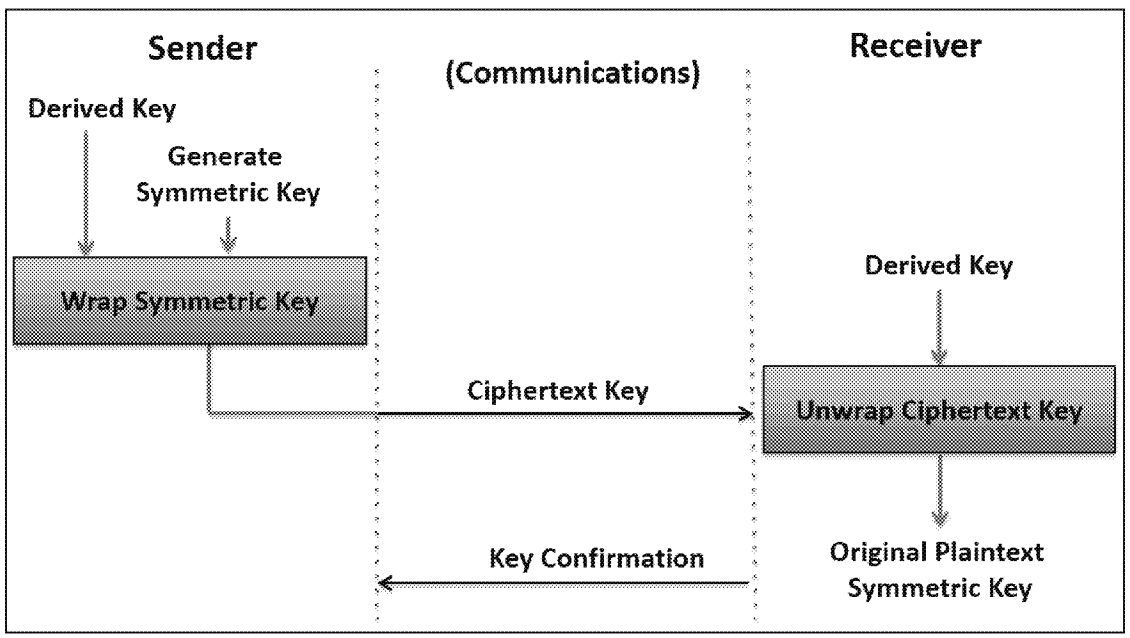

Figure 7: SP 800-56A Key Transport Example

Figure 7 illustrates the key transport process that follows the key-agreement discussed in Section 5.3.3 and shown in Figure 6. After the key-agreement part of the transaction, the initiator and responder share a symmetric key-wrapping key, which is then used as follows:

The sender:

1. Generates (or otherwise obtains) a symmetric key to be transported (note that the sender could have been either the initiator or the responder in the key-agreement part of the transaction),
2. Wraps the symmetric key from step 1 using the key-wrapping key, and
3. Sends the resulting ciphertext (i.e., the wrapped key) to the intended receiver.

The receiver:

4. Unwraps the ciphertext using his copy of the key-wrapping key to obtain the original plaintext symmetric key, and
5. Optionally performs key confirmation; although this step is optional, it is highly recommended to provide assurance that both parties now have the same symmetric key.

5.3.4.2 SP 800-56B Key Transport

SP 800-56B specifies two very different methods for transporting keys whereby the sender uses the receiver's public key to securely transport keying material to the receiver.

Figure 8 provides a simplified example of one of the key-transport methods in SP 800-56B. In both methods, the receiver must have a key pair that is used during a key-transport transaction. In the example shown in the figure, key transport is accomplished as follows.

The sender:

1. Obtains the public key of the intended receiver,
2. Generates a symmetric key to be transported,
3. Encrypts the symmetric key using the receiver's public key, and
4. Sends the resulting ciphertext key to the receiver.

The receiver:

5. Uses his private key to decrypt the ciphertext key, thus obtaining the original plaintext key.
6. Optionally performs key confirmation; although this step is optional, it is highly recommended to provide assurance that both parties now have the same symmetric key.

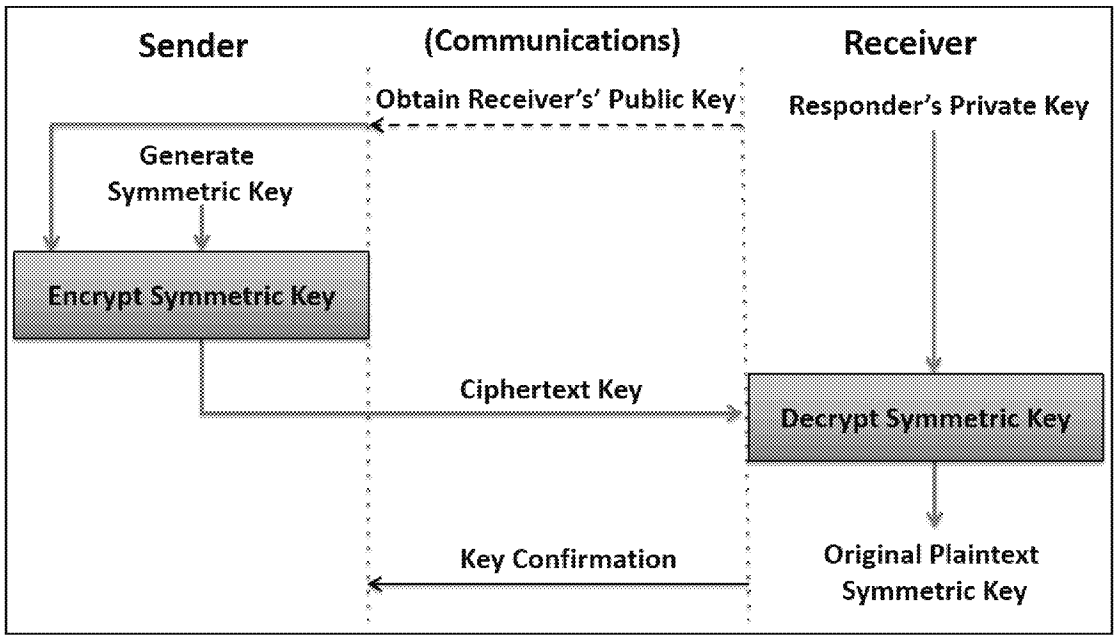

Figure 8: SP 800-56B Key Transport Example

5.3.5 Key Wrapping

Key wrapping is a method used to provide confidentiality and integrity protection to keys (and possibly other information) using a symmetric key-wrapping key that is known by both the sender and receiver, and a symmetric-key block cipher algorithm. The wrapped keying material can then be stored or transmitted (i.e., transported) securely. Unwrapping the keying material requires the use of the same algorithm and key-wrapping key that was used during the original wrapping process.

Key wrapping differs from simple encryption in that the wrapping process includes an integrity feature. During the unwrapping process, this integrity feature is used to detect accidental or intentional modifications to the wrapped keying material.

Three methods have been specified in SP 800-38F[74] for key wrapping, and other SP 800-38 modes (or combination of modes) that that can also be used for key wrapping are also **approved** in SP 800-38F. Depending on the method or mode, either AES or TDEA can be used.

5.3.6 Derivation of a Key from a Password

Keys can be derived from passwords. Due to the ease of guessing most passwords, keys derived in this manner are not suitable to be used for most applications. However, SP 800-132[75] specifies a family of functions that can be used to derive keying material from

[74] SP 800-38F, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping.*

[75] SP 800-132, *Recommendation for Password-Based Key Derivation Part 1: Storage Applications.*

a password[76] for electronic storage applications (e.g., when encrypting an entire disk drive).

5.4 Key Management Issues

A number of issues need to be addressed for selecting and using a CKMS.

5.4.1 Manual vs. Automated Key Establishment

As discussed in Sections 5.3 and 5.3.4, keys can be established between entities either manually or using automated methods. In many cases, a hybrid approach is used in which an entity generates and manually distributes one or more keys to other entities, and thereafter these keys are used to establish other keys (see SP 800-56A and SP 800-56B).

The number of keys to be manually distributed depends on the type of cryptography to be used (i.e., symmetric or asymmetric methods) and must be considered when selecting the capabilities required of a CKMS.

5.4.2 Selecting and Operating a CKMS

A CKMS could be designed, implemented and operated by the organization that will use it. Or, the organization could operate a CKMS procured from a vendor. Or, an organization could procure the services of a third party that procures a CKMS from a vendor. Whichever choice is made, the organization needs to make sure that the CKMS that is used provides the protections that are required for the organization's information. SP 800-130 and SP 800-152 discuss the considerations that need to be addressed by the federal organization, including the scalability of the CKMS, and the metadata to be associated with the keys.

5.4.3 Storing and Protecting Keys

Keys can be stored in a number of places and protected in a variety of ways. They could be stored in a safe. They could be present only in a validated cryptographic module where the module itself might adequately protect the keys, depending on its design. Keys could also be stored on electronic media, such as a flash drive; in this case, a key may need to be encrypted or split into key components so that no single person can determine what the key is. These issues need to be addressed for operational keys.

Certain keys may need to be backed up so that if an operational key is inadvertently lost or modified, it can be recovered and operations resumed. Some keys may also need to be archived for long-term storage (e.g., because of legal requirements or to decrypt archived data). A key-recovery capability is needed whenever keys are backed up or archived. This capability needs to be designed so that the keys can be recovered in an acceptable amount of time and only by those entities authorized to do so; see SP 800-57, Part 1 for more information about key backup, key archiving and key recovery.

[76] Note that this publication considers a passphrase to be a password.

5.4.4 Cryptoperiods

A cryptoperiod is the time span during which a specific key is authorized for use. A cryptoperiod for a key is assigned for a number of reasons, including limiting the amount of exposure of encrypted data if a single key is compromised. Cryptoperiods are usually assigned for a carefully considered period of time or by the maximum amount of data protected by the key. Tradeoffs associated with the determination of a cryptoperiod involve the risks and consequences of exposure. Section 5.3 of SP 800-57, Part 1 provides a more detailed discussion of the need for establishing cryptoperiods, the factors to be considered when deciding on a suitable cryptoperiod and some suggestions for the length of cryptoperiods.

5.4.5 Use Validated Algorithms and Cryptographic Modules

Cryptographic algorithms must be validated and implemented in FIPS 140-validated cryptographic modules. Every IT product available makes a claim as to functionality and/or offered security. When protecting sensitive data, a minimum level of assurance is needed that a product's stated security claim is valid. There are also legislative restrictions regarding certain types of technology, such as cryptography, that require federal agencies to use only tested and validated products.

Federal agencies, private industry, and the public rely on cryptography for the protection of information and communications used in electronic commerce, the critical infrastructure, and other application areas. At the core of all products offering cryptographic services is the cryptographic module. Cryptographic modules, which contain cryptographic algorithms, are used in products and systems to provide security services such as confidentiality, integrity, and authentication. Although cryptography is used to provide security, weaknesses such as poor design or weak algorithms can render the product insecure and place highly sensitive information at risk. Adequate testing and validation of the cryptographic module and its underlying cryptographic algorithms against established standards is essential to provide security assurance.

NIST has established programs to validate the implementation of the **approved** cryptographic algorithms and the cryptographic modules in which they are used: the Cryptographic Algorithm Validation Program (CAVP) and the Cryptographic Module Validation Program (CMVP). Information about the CAVP is available at http://csrc.nist.gov/groups/STM/cavp/, while information about the CMVP is available at http://csrc.nist.gov/groups/STM/cmvp/.

Also, see Section 5.1.2 in this document for a discussion of the security requirements for cryptographic modules.

5.4.6 Control of Keying Material

The access to keys needs to be controlled. A key should only be accessible by an authorized entity, and only for the purpose for which it is authorized. For example, a key

designated for key transport must not be used for the generation or verification of digital signatures.

The proliferation of keys also needs to be controlled. While it is often convenient to make copies of keys, these extra copies need to be accounted for. If a key is compromised, that key and all its copies may need to be destroyed to prevent subsequent unauthorized use. For example, if a private key used for the generation of a digital signature is compromised, and a copy of the key still exists after the original copy was destroyed, then there is a possibility that the copy could be used to generate unauthorized digital signatures at a later time.

Users must be provided with a list of responsibilities and liabilities, and each user should sign a statement acknowledging these concerns before receiving a key. Users must be made aware of their unique responsibilities, especially regarding the significance of a key compromise or loss. Users must be able to store their secret and private keys securely, so that no intruder can access them, yet the keys must be readily accessible for legitimate use.

5.4.7 Compromises

It is imperative to have a plan for handling the compromise or suspected compromise of keys, particularly those used and managed at a central site (e.g., the keys used by a CA to sign certificates); this should be established before the system becomes operational. A compromise-recovery plan should address what actions will be taken with compromised system software and hardware, CA keys, user keys, previously generated signatures, encrypted data, etc. SP 800-57, Part 1 includes discussions of the effects of a key compromise, measures for minimizing the likelihood or consequences of a key compromise, and what should be considered in developing a compromise-recovery plan.

If someone's private or secret key is lost or compromised, other users must be made aware of this, so that they will no longer initiate the protection of data using a compromised key, or accept data protected with a compromised key without assessing and accepting the risk of doing so. This notification is often accomplished using CRLs or Compromised Key Lists (CKLs); see SP 800-57, Part 1 for discussions.

In some cases, a key and all copies of the key should be destroyed immediately upon the detection of a key compromise. For example, a private key used for the generation of digital signatures should be immediately destroyed. However, the corresponding public key may need to remain available for verifying the signatures that were previously generated using the compromised private key. Note that there is a risk associated with accepting these signatures.

5.4.8 Accountability and Auditing

Accountability involves the identification of those entities that have access to or control of cryptographic keys throughout their lifecycles. Accountability can be an effective tool to help prevent key compromises and to reduce the impact of compromises when they are detected. Accountability 1) aids in the determination of when a compromise could have occurred and what individuals could have been involved, 2) discourages key compromise because users know their access to the key is known, and 3) is useful in determining

where the key was used and what data or other keys were protected by a compromised key, and therefore, may also be compromised.

Auditing is another mechanism used for the detection of and recovery from key compromises. Auditing includes reviewing the actions of humans that use, operate and maintain systems, looking for unusual events that may indicate inappropriate actions by the humans or processes using a key management system.

SECTION 6: OTHER ISSUES

The use of cryptography should not be undertaken without a thorough risk analysis, and a determination of the sensitivity of the information to be protected and the security controls to be used (see SP 800-175A and SP 800-53). After performing a risk assessment and determining the sensitivity level of the information to be protected (Low, Moderate or High) and the security controls to be used, a number of issues need to be addressed to ensure that cryptography is used properly.

This section identifies issues to be addressed after determining that cryptography is required.

6.1 Required Security Strength

The minimum security strength is determined by the sensitivity level of the information (see FIPS 199). SP 800-152 requires a security strength of at least 112 bits for the protection of Low-impact information, 128 bits for Moderate-impact information, and 192 bits for High-impact information. The required security strength can then be used to determine the algorithm and key size to be used. Section 5.6 of SP 800-57, Part 1 provides tables for selecting appropriate algorithms and key sizes.

Many applications require the use of several different cryptographic algorithms. Ideally, these algorithms would all offer the same security strength, but this may not always be the case for performance, availability and interoperability reasons. When algorithms of different strengths are used together to protect data, the security provided by the combination of algorithms is the strength associated with the algorithm with the lowest security strength (see Section 5.6 of SP 800-57, Part 1). For example, RSA with 2048-bit keys can support a security strength of 112 bits, but is often used with SHA-256, which can support a security strength of 128 bits. When the combination is used to generate a digital signature, the signature can only provide a security strength of 112 bits – the lesser strength offered by the two algorithms.

Approved combinations of algorithms (called cipher suites) for some of the protocols are provided in SP 800-57, Part 3 (for S/MIME) and SP 800-52 (for TLS).

6.2 Interoperability

Interoperability is the ability of one entity to communicate with another entity, whether the entities are people, devices or processes. In order to communicate, the entities must have:

- A communications channel (e.g., the Internet) and the same communications protocol (e.g., TLS), and
- Policies that allow the entities to communicate.

In order to communicate securely, the entities must also have:

- Trust that each entity will enforce its own policies.

- Interoperable cryptographic capabilities as discussed in Section 4, and
- Share appropriate keying material that has been established securely (see Section 5.3).

For example, if entities A and B are in two different organizations, and

- Each organization has a policy that allows the entities to communicate,
- Each entity trusts that the other entity will enforce its own policies,
- There is a TLS capability that can be used for communication,
- Each entity can encrypt and decrypt information using AES with a 128-bit key and establish keys using 3072-bit RSA key transport (see Section 5.3.4), and
- One of the entities can generate a 128-bit AES key and act as the sender in the key-transport scheme, and the other entity has a 3072-bit RSA key pair and can act as the receiver (see Section 5.3.4.2 for a discussion on key transport),

then the two entities have a secure and interoperable communication channel that can be used to establish a 128-bit key for encrypting information using AES. In this case, the security strength that can be provided by an encryption operation using AES is 128 bits, since both 3072-bit RSA and AES-128 are rated at a security strength of 128 bits (see Section 6.1).

6.3 When Algorithms are No Longer Approved

In the case that an algorithm is **no longer approved** for providing adequate protection (e.g., the algorithm may have been "broken"), a risk assessment needs to be performed to determine whether the information should be re-protected using an **approved** algorithm and key size that will protect the information for the remainder of its security life. See Section 5.6.4 for SP 800-57, Part 1 for additional discussion.

6.4 Registration Authorities (RAs)

As discussed in Section 5.2.3.1, an RA verifies the identity of users applying for a certificate and authenticates other information to be included in a certificate generated by a Certification Authority (CA). The correctness of this information is the linchpin on which the security of using certificates is based. Once this information is verified, the appropriate information is submitted to a CA for certificate generation using a signed certification request. The CA must deem the RA as trustworthy, e.g.,

- Appropriate identification is provided by an entity requesting a certificate and is fully checked by the RA;
- Information submitted for inclusion in the certificate is checked for validity (e.g., that the public key is valid, and the private key is in the possession of the claimed owner); and
- The RA provides adequate protection for the private key used to sign the certification request.

6.5 Cross Certification

Cross certification is the establishment of a trust relationship between two Certification Authorities (CAs) through the signing of each other's public key in a certificate referred to as a "cross-certificate." Cross-certificates provide a means to create a chain of trust from a single, trusted, root CA to multiple other CAs so that subscribers in one CA domain can interact safely with subscribers in other CA domains (e.g., the subscriber in one CA domain has assurance of the identity of the subscriber in the other domain and assurance of the accurateness of the other information provided by his certificate).

Cross certification should only be performed when each CA examines the other CA's policies, finds them acceptable and trusts that CA to operate in accordance with those policies.

Appendix A: References

The following FIPS and NIST Special Publications (SP) apply to the use of cryptography in the Federal Government.

All publications are available at http://csrc.nist.gov/publications.

FIPS 140	Federal Information Processing Standard 140-2, *Security Requirements for Cryptographic Modules*, May 25, 2001 (updated December 3, 2002 (Change Notice 2)). http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf [accessed 8/18/16]. FIPS 140-2 specifies the requirements that must be met by cryptographic modules protecting U.S. Government information. The standard provides four increasing, qualitative levels of security. The security requirements cover areas related to the secure design and implementation of a cryptographic module.
FIPS 180	Federal Information Processing Standard 180-4, *Secure Hash Standard (SHS)*, August 2015. http://dx.doi.org/10.6028/NIST.FIPS.180-4 FIPS 180-4 specifies seven cryptographic hash algorithms: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256.
FIPS 185	Federal Information Processing Standard 185, *Escrowed Encryption Standard*, February 9, 1994 [withdrawn October 19, 2015]. http://csrc.nist.gov/publications/fips/fips185/fips185.pdf [accessed 8/18/16]. FIPS 185 specified the use of an encryption/decryption algorithm and a Law Enforcement Access Field (LEAF) creation method that could be implemented in electronic devices and used for protecting government telecommunications when such protection was desired. The algorithm and the LEAF creation method were classified. The LEAF was intended for use in a key escrow system that provided for the decryption of telecommunications when access to the telecommunications was lawfully authorized.
FIPS 186	Federal Information Processing Standard 186-4, *Digital Signature Standard (DSS)*, July 2013. http://dx.doi.org/10.6028/NIST.FIPS.180-4 FIPS 186-4 specifies a suite of algorithms that can be used to generate a digital signature: DSA, ECDSA and RSA. This Standard includes methods for the generation of digital signatures, methods for the generation of domain parameters (for DSA and ECDSA), and methods for the generation of key pairs, and requires certain

	assurances for using digital signatures: assurance of domain-parameter validity (DSA and ECDSA), and assurance of public-key validity and assurance of private-key possession for all three algorithms.
FIPS 197	Federal Information Processing Standard 197, *Advanced Encryption Standard (AES)*, November 26, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf [accessed 8/18/16]. FIPS 197 specifies a symmetric key block cipher algorithm. The Standard supports key sizes of 128, 192, and 256 bits and a block size of 128 bits.
FIPS 198	Federal Information Processing Standard 198-1, *Keyed-Hash Message Authentication Code (HMAC)*, July 2008. http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf [accessed 8/18/16]. FIPS 198-1 defines a message authentication code (MAC) that uses a cryptographic hash function in conjunction with a secret key for the calculation and verification of the MACs.
FIPS 199	Federal Information Processing Standard 199, *Standards for Security Categorization of Federal Information and Information Systems*, February 2004. http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf [accessed 8/18/16]. FIPS 199 establishes security categories for both information and information systems. The security categories are based on the potential impact on an organization if certain events occur that jeopardize the information and information systems needed by the organization to accomplish its assigned mission, protect its assets, fulfill its legal responsibilities, maintain its day-to-day functions, and protect individuals.
FIPS 202	Federal Information Processing Standard 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, August 2015. http://dx.doi.org/10.6028/NIST.FIPS.202 FIPS 202 specifies SHA3-224, SHA3-256, SHA3-384 and SHA3-512. This FIPS also specifies two extendable-output functions (SHAKE128 and SHAKE256), which are not, in themselves, considered to be hash functions.

SP 800-22	Special Publication 800-22 Revision 1a, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, April 2010. http://dx.doi.org/10.6028/NIST.SP.800-22r1a SP 800-22 discusses some aspects of selecting and testing random and pseudorandom number generators for providing random numbers that are indistinguishable from truly random output.
SP 800-32	Special Publication 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure* , February 26, 2001. http://dx.doi.org/10.6028/NIST.SP.800-32 SP 800-32 was developed to assist agency decision-makers in determining if a PKI is appropriate for their agency, and how PKI services can be deployed most effectively within a Federal agency. It is intended to provide an overview of PKI functions and their applications.
SP 800-38	A series of publications specifying modes of operation for block cipher algorithms.
SP 800-38A	Special Publication 800-38A, *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*, December 2001. http://dx.doi.org/10.6028/NIST.SP.800-38A SP 800-38A defines five confidentiality modes of operation for use with an underlying symmetric key block cipher algorithm: Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). Used with an **approved** underlying block cipher algorithm (i.e., AES and TDEA), these modes can provide cryptographic protection for sensitive computer data.
SP 800-38B	Special Publication 800-38B, *Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication*, May 2005. http://dx.doi.org/10.6028/NIST.SP.800-38B SP 800-38B specifies a message authentication code (MAC) algorithm based on a symmetric key block cipher (i.e., AES or TDEA). This block cipher-based MAC algorithm, called CMAC, may be used to provide assurance of the source and integrity of binary data.

SP 800-38C	Special Publication 800-38C, *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality*, May 2004 (updated July 20, 2007). http://dx.doi.org/10.6028/NIST.SP.800-38C SP 800-38C defines a mode of operation, called CCM, for a symmetric-key block cipher algorithm with a 128-bit block size (i.e., AES). CCM may be used to provide assurance of the confidentiality and the authenticity of computer data by combining the techniques of the Counter (CTR) mode specified in SP 800-38A, and the Cipher Block Chaining-Message Authentication Code (CBC-MAC) algorithm (specified in SP 800-90B, but **not** currently approved for general use).
SP 800-38D	Special Publication 800-38D, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, November 2007. http://dx.doi.org/10.6028/NIST.SP.800-38D SP 800-38D specifies the Galois/Counter Mode (GCM), an algorithm for authenticated encryption with associated data, and its specialization, GMAC, for generating a message authentication code (MAC) on data that is not encrypted. GCM and GMAC are modes of operation for an underlying, **approved** symmetric-key block cipher with a 128-bit block size (i.e., AES).
SP 800-38E	Special Publication 800-38E, *Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices*, January 2010. http://dx.doi.org/10.6028/NIST.SP.800-38E SP 800-38E approves the XTS-AES mode of the AES algorithm by reference to IEEE 1619, subject to one additional requirement, as an option for protecting the confidentiality of data on storage devices. The mode does not provide authentication of the data or its source.

SP 800-38F	Special Publication 800-38F, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*, December 2012. http://dx.doi.org/10.6028/NIST.SP.800-38F SP 800-38F describes cryptographic methods that are approved for key wrapping. In addition to approving existing methods, this publication specifies two new, deterministic authenticated-encryption modes of operation of the Advanced Encryption Standard (AES) algorithm: the AES Key Wrap (KW) mode and the AES Key Wrap with Padding (KWP) mode. An analogous mode with the Triple Data Encryption Algorithm (TDEA) as the underlying block cipher, called TKW, is also specified to support legacy applications.
SP 800-38G	Special Publication 800-38G, *Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*, March 2016. http://dx.doi.org/10.6028/NIST.SP.800-38G SP 800-38G specifies methods for format-preserving encryption, called FF1 and FF3. Each of these methods is a mode of operation of the AES algorithm, which is used to construct a round function within the Feistel structure for encryption.
SP 800-52	Special Publication 800-52 Revision 1, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*, April 2014. http://dx.doi.org/10.6028/NIST.SP.800-52r1 Transport Layer Security (TLS) provides mechanisms to protect sensitive data during electronic dissemination across the Internet. SP 800-52 provides guidance about the selection and configuration of TLS protocol implementations, while making effective use of Federal Information Processing Standards (FIPS) and NIST-recommended cryptographic algorithms (specified in SPs), and requires that TLS 1.1 be configured with FIPS-based cipher suites as the minimum appropriate secure transport protocol. This publication also identifies TLS extensions for which mandatory support must be provided and identifies other recommended extensions.
SP 800-53	Special Publication 800-53 Revision 4, *Security and Privacy Controls for Federal Information Systems and Organizations*, April 2013 (updated January 22, 2015). http://dx.doi.org/10.6028/NIST.SP.800-53r4 SP 800-53 provides a catalog of security and privacy controls for federal information systems and organizations, and a process for selecting controls to protect organizational operations (including

	mission, functions, image, and reputation), organizational assets, individuals, other organizations, and the Nation from a diverse set of threats, including hostile cyber attacks, natural disasters, structural failures, and human errors.
SP 800-56A	Special Publication 800-56A Revision 2, *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography*, May 2013. http://dx.doi.org/10.6028/NIST.SP.800-56Ar2 SP 800-56A specifies key-establishment schemes based on the discrete logarithm problem over finite fields and elliptic curves, including several variations of Diffie-Hellman and Menezes-Qu-Vanstone (MQV) key-establishment schemes.
SP 800-56B	Special Publication 800-56B Revision 1, *Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography*, September 2014. http://dx.doi.org/10.6028/NIST.SP.800-56Br1 SP 800-56B specifies key-establishment schemes using integer-factorization cryptography (RSA). Both key transport and key-agreement schemes are specified.
SP 800-56C	Special Publication 800-56C, *Recommendation for Key Derivation through Extraction-then-Expansion*, November 2011. http://dx.doi.org/10.6028/NIST.SP.800-56C SP 800-56C specifies techniques for the derivation of keying material from a shared secret established during a key-establishment scheme defined in SP 800-56A or SP 800-56B through an extraction-then-expansion procedure.
SP 800-57, Part 1	Special Publication 800-57, Part 1 Revision 4, *Recommendation for Key Management, Part 1: General*, January 2016. http://dx.doi.org/10.6028/NIST.SP.800-57pt1r4 Part 1 of SP 800-57 provides general guidance and best practices for the management of cryptographic keying material. It focuses on issues involving the management of cryptographic keys: their generation, use, and eventual destruction. Related topics, such as algorithm selection and appropriate key size, cryptographic policy, and cryptographic module selection, are also included.

SP 800-57, Part 2	Special Publication 800-57, Part 2, *Recommendation for Key Management, Part 2: Best Practices for Key Management Organization*, August 2005. http://dx.doi.org/10.6028/NIST.SP.800-57p2 Part 2 of SP 800-57 provides guidance on policy and security planning requirements for U.S. government agencies. This part of SP 800-57 contains a generic key-management infrastructure, guidance for the development of organizational key-management policy statements and key-management practices statements, an identification of key-management information that needs to be incorporated into security plans for general support systems and major applications that employ cryptography, and an identification of key-management information that needs to be documented for all Federal applications of cryptography.
SP 800-57, Part 3	Special Publication 800-57, Part 3 Revision 1, *Recommendation for Key Management, Part 3: Application-Specific Key Management Guidance*, January 2015. http://dx.doi.org/10.6028/NIST.SP.800-57pt3r1 Part 3 of SP 800-57 addresses the key-management issues associated with currently available cryptographic mechanisms, such as the Public Key infrastructure (PKI), Internet Protocol Security (IPsec), the Transport Layer Security protocol (TLS), Secure/Multipart Internet Mail Extensions (S/MIME), Kerberos, Over-the-Air Rekeying (OTAR), Domain Name System Security Extensions (DNSSEC), Encrypted File Systems and the Secure Shell (SSH) protocol.
SP 800-67	Special Publication 800-67 Revision 1, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, January 2012. http://dx.doi.org/10.6028/NIST.SP.800-67r1 SP 800-67 specifies the Triple Data Encryption Algorithm (TDEA), including its primary component cryptographic engine, the Data Encryption Algorithm (DEA).
SP 800-89	Special Publication 800-89, *Recommendation for Obtaining Assurances for Digital Signature Applications*, November 2006. http://dx.doi.org/10.6028/NIST.SP.800-89 Entities participating in the generation or verification of digital signatures depend on the authenticity of the process. SP 800-89 specifies methods for obtaining the assurances necessary for valid digital signatures: assurance of domain parameter validity, assurance

	of public key validity, assurance that the key-pair owner actually possesses the private key, and assurance of the identity of the key pair owner.
SP 800-90A	Special Publication 800-90A Revision 1, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, June 2015. http://dx.doi.org/10.6028/NIST.SP.800-90Ar1 SP 800-90A specifies DRBG mechanisms for the generation of random bits using deterministic methods. The methods provided are based on either hash functions or block cipher algorithms and are designed to support selected security strengths. DRBGs must be initialized from a randomness source that provides sufficient entropy for the security strength to be supported by the DRBG.
SP 800-90B	Special Publication 800-90B (Draft), *Recommendation for the Entropy Sources Used for Random Bit Generation*, January 2016. http://csrc.nist.gov/publications/PubsSPs.html#800-90B [accessed 8/18/16]. SP 800-90B specifies the design principles and requirements for the entropy sources used by Random Bit Generators, including health tests to determine that the entropy source has not failed and tests for the validation of entropy sources.
SP 800-90C	Special Publication 800-90C (Draft), *Recommendation for Random Bit Generator (RBG) Constructions*, April 2016. http://csrc.nist.gov/publications/PubsSPs.html#800-90C [accessed 8/18/16]. SP 800-90C specifies constructions for the implementation of random bit generators (RBGs). An RBG may be a deterministic random bit generator (DRBG) or a non-deterministic random bit generator (NRBG). The constructed RBGs consist of DRBG mechanisms as specified SP 800-90A and entropy sources as specified in SP 800-90B.

SP 800-102	Special Publication 800-102, *Recommendation for Digital Signature Timeliness*, September 2009. http://dx.doi.org/10.6028/NIST.SP.800-102 Establishing the time when a digital signature was generated is often a critical consideration. A signed message that includes the (purported) signing time provides no assurance that the private key was used to sign the message at that time unless the accuracy of the time can be trusted. With the appropriate use of digital signature-based timestamps from a Trusted Timestamp Authority and/or verifier-supplied data that is included in the signed message, the signer can provide some level of assurance about the time that the message was signed.
SP 800-106	Special Publication 800-106, *Randomized Hashing for Digital Signatures*, February 2009. http://dx.doi.org/10.6028/NIST.SP.800-106 NIST-approved digital signature algorithms require the use of an **approved** cryptographic hash function in the generation and verification of signatures. SP 800-106 specifies a method to enhance the security of the cryptographic hash functions used in digital signature applications by randomizing the messages that are signed.
SP 800-107	Special Publication 800-107 Revision 1, *Recommendation for Applications Using Approved Hash Algorithms*, August 2012. http://dx.doi.org/10.6028/NIST.SP.800-107r1 Hash functions that compute a fixed-length message digest from arbitrary length messages are widely used for many purposes in information security. SP 800-107 provides security guidelines for achieving the required or desired security strengths when using cryptographic applications that employ the approved hash functions specified in FIPS 180. These include functions such as digital signatures, Keyed-hash Message Authentication Codes (HMACs) and Hashed-based Key Derivation Functions (hash-based KDFs).
SP 800-108	Special Publication 800-108, *Recommendation for Key Derivation Using Pseudorandom Functions (Revised)*, October 2009. http://dx.doi.org/10.6028/NIST.SP.800-108 SP 800-108 specifies techniques for the derivation of additional keying material from a secret key (i.e., a key-derivation key) using pseudorandom functions. The key-derivation key may have been either established through a key-establishment scheme or shared through some other manner (e.g., a manual key distribution).

SP 800-130	Special Publication 800-130, *A Framework for Designing Cryptographic Key Management Systems*, August 2013. http://dx.doi.org/10.6028/NIST.SP.800-130 SP 800-130 contains topics to be considered by a CKMS designer when developing a CKMS design specification. Topics include security policies, cryptographic keys and metadata, interoperability and transitioning, security controls, testing and system assurances, disaster recovery, and security assessments.
SP 800-131A	Special Publication 800-131A Revision 1, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, November 2015. http://dx.doi.org/10.6028/NIST.SP.800-131Ar1 Section 5.6.4 of SP 800-57, Part 1 provides recommendations for transitioning to new cryptographic algorithms and key lengths because of algorithm breaks or the availability of more powerful computers that could be used to efficiently search for cryptographic keys. SP 800-131A offers more specific guidance for such transitions. Each algorithm and service is addressed in SP 800-131A, indicating whether its use is acceptable, deprecated, restricted, allowed only for legacy applications[77], or disallowed.
SP 800-132	Special Publication 800-132, *Recommendation for Password-Based Key Derivation, Part 1: Storage Applications*, December 2010. http://dx.doi.org/10.6028/NIST.SP.800-132 SP 800-132 specifies techniques for the derivation of master keys from passwords or passphrases to protect stored electronic data or data protection keys.
SP 800-133	Special Publication 800-133, *Recommendation for Cryptographic Key Generation*, December 2012. http://dx.doi.org/10.6028/NIST.SP.800-133 SP 800-133 discusses the generation of the keys to be managed and used by the **approved** cryptographic algorithms.

[77] The algorithm and key length may be used to process already-protected information, but there may be a risk in doing so.

SP 800-135	Special Publication 800-135 Revision 1, *Recommendation for Existing Application-Specific Key Derivation Functions*, December 2011. http://dx.doi.org/10.6028/NIST.SP.800-135r1 Many widely-used internet security protocols have their own application-specific Key Derivation Functions (KDFs) that are used to generate the cryptographic keys required for their cryptographic functions. SP 800-135 provides security requirements for those KDFs.
SP 800-152	Special Publication 800-152, *A Profile for U. S. Federal Cryptographic Key Management Systems (CKMS)*, October 2015. http://dx.doi.org/10.6028/NIST.SP.800-152 SP 800-152 contains requirements for the design, implementation, procurement, installation, configuration, management, operation and use of a CKMS by and for U.S. federal organizations and their contractors. The Profile is based on NIST Special Publication SP 800-130.
SP 800-175A	Special Publication 800-175A, *Guideline for Using Cryptographic Standards in the Federal Government: Directives, Mandates and Policies*, August 2016. http://dx.doi.org/10.6028/NIST.SP.800-175A SP 800-175A provides guidance on the determination of requirements for using cryptography. It includes a summary of laws and regulations concerning the protection of the Federal Government's sensitive information, guidance regarding the conduct of risk assessments to determine what needs to be protected and how best to protect that information, and a discussion of the relevant security-related documents (e.g., various policy and practice documents).
NISTIR 7924	NIST Internal Report 7924 (Second Draft), *Reference Security Policy*, May 2014. http://csrc.nist.gov/publications/PubsDrafts.html#NIST-IR-7924 [accessed 8/18/16]. NIST 7924 is intended to identify a set of security controls and practices to support the secure issuance of certificates. It was written in the form of a Certificate Policy (CP), a standard format for defining the expectations and requirements of the relying party community that will trust the certificates issued by its Certificate Authorities (CAs).

Non-NIST Publications:

IEEE 802.11	*Wireless Local Area Networks* [web page]. http://standards.ieee.org/about/get/802/802.11.html
IEEE P1363	IEEE P1363: *Standard Specifications for Public-Key Cryptography* [web page]. http://grouper.ieee.org/groups/1363/
IEEE P1363a	IEEE P1363a: *Standard Specifications For Public Key Cryptography-Amendment 1: Additional Techniques*, 2004.
IEEE P1363.1	*Public-Key Cryptographic Techniques Based on Hard Problems over Lattices*, October 2008.
IEEE P1363.2	*Password-Based Public-Key Cryptography*, 2008.
IEEE P1619	*Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices*, 2008.
ISO/IEC 9594-8	ITU-T Recommendation X.509 (2012) | ISO/IEC 9594-8:2014, *Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*.
ISO/IEC 9797-1	ISO/IEC 9797-1:2011, *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher*, March 2011. This standard includes CMAC, as specified in SP 800-38B.
ISO/IEC 9797-2	ISO/IEC 9797-2:2011, *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function*, May 2011. This standard includes HMAC, as specified in FIPS 198.
ISO/IEC 10116	ISO/IEC 10116:2006, *Information technology – Security techniques – Modes of operation for an n-bit block cipher*, February 2006. This standard includes all the modes specified in SP 800-38A.
ISO/IEC 10118-3	ISO/IEC 10118-3:2004, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*, March 2004. This standard includes SHA-1 and the SHA-2 family of hash functions specified in FIPS 180. A revision of ISO/IEC 10118-3 will

	include the SHA-3 functions specified in FIPS 202.
ISO/IEC 11770-3	ISO/IEC 11770-3: 2015, *Information technology – Security techniques – Key management -- Part 3: Mechanisms using asymmetric techniques*, August 2015. This standard specifies key establishment mechanisms, some of which can be instantiated with key-establishment schemes specified in SP 800-56A and SP 800-56B.
ISO/IEC FDIS 11770-6	ISO/IEC FDIS 11770-6, *Information technology – Security techniques – Key management – Part 6: Key derivation*, 2015. This draft standard will include all key derivation functions specified in SP 800-108, as well as the two-step key derivation methods specified in SP 800-56C.
ISO/IEC 11889	ISO/IEC 11889-1:2015, *Information technology – Trusted Platform Module Library – Part 1: Architecture*, August 2015. ISO/IEC 11889-2:2015, *Information technology – Trusted Platform Module – Part 2: Structures*, August 2015. ISO/IEC 11889-3:2015, *Information technology – Trusted Platform Module – Part 3: Commands*, August 2015. ISO/IEC 11889-4:2015, *Information technology – Trusted Platform Module Library – Part 4: Supporting Routines*, August 2015.
ISO/IEC 14888-2	ISO/IEC 14888-2:2008, *Information technology – Security techniques – Digital signatures with appendix – Part 2: Integer factorization based mechanisms*, April 2008. This standard includes RSA signatures, as specified in FIPS 186.
ISO/IEC 14888-3	ISO/IEC 14888-3:2016, *Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms*, March 2016. This standard includes DSA, as specified for finite fields and elliptic curves in FIPS 186.
ISO/IEC 18033-3	ISO/IEC 18033-3:2010, *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers*, December 2010. This standard includes 64-bit block ciphers: TDEA, MISTY1, CAST-128, HIGHT and 128-bit block ciphers: AES, Camellia, and SEED. Note that TDEA is specified in SP 800-67 and AES is specified in FIPS 197.

ISO/IEC 19772	ISO/IEC 19772:2009, *Information technology – Security techniques – Authenticated encryption*, February 2009. This standard includes CCM (as specified in SP 800-38C), GCM (as specified in SP 800-38D), and Key wrapping (as specified in SP 800-38E).
PKCS 1	Public Key Cryptography System #1, version 2.2, *RSA Cryptography Standard*, October 27, 2012. http://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11300-wp-pkcs-1v2-2-rsa-cryptography-standard.pdf [accessed 8/18/16]. PKCS 1 provides recommendations for the implementation of public-key cryptography based on the RSA algorithm, covering cryptographic primitives, encryption schemes, signature schemes with appendix and the ASN.1 syntax for representing keys and for identifying the schemes.
X9.31	American National Standard for Financial Services X9.31-1998, *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*, 1998 [withdrawn]. ANS X9.31 defined a method for digital signature (signature) generation and verification for the protection of financial messages and data using reversible public key cryptography systems without message recovery. In addition, criteria for the generation of public and private keys required by the algorithm and the procedural controls required for the secure use of the algorithm were provided.
X9.42	American National Standard for Financial Services X9.42-2001, *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, 2001 [withdrawn]. ANS X9.42, partially adapted from ISO 11770-3, specifies schemes for the agreement of symmetric keys using the Diffie-Hellman and MQV algorithms. It covers methods for domain-parameter generation, domain-parameter validation, key-pair generation, public-key validation, shared secret value calculation, key derivation, and test message authentication code computation for discrete-logarithm problem-based key-agreement schemes.

X9.44	American National Standard for Financial Services X9.44-2007, *Key Establishment Using Integer Factorization Cryptography*, 2007. ANS X9.44 specifies key-establishment schemes using public-key cryptography, based on the integer factorization problem. Two types of key-establishment schemes are specified: key transport and key agreement.
X9.62	American National Standard X9.62-2005, *Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA)*, 2005. ANS X9.62 defines methods for digital signature (signature) generation and verification for the protection of messages and data using the Elliptic Curve Digital Signature Algorithm (ECDSA). This Standard provides methods and criteria for the generation of public and private keys that are required by ECDSA and the procedural controls required for the secure use of the algorithm with these keys. This ECDSA Standard also provides methods and criteria for the generation of elliptic-curve domain parameters that are required by ECDSA and the procedural controls required for the secure use of the algorithm with these domain parameters.
X9.63	American National Standard X9.63-2011, *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*, 2011. ANS X9.63 defines key-establishment schemes that employ asymmetric cryptographic techniques. The arithmetic operations involved in the operation of the schemes take place in the algebraic structure of an elliptic curve over a finite field. Both key-agreement and key-transport schemes are specified.